

Tecnologie Web T
11 Settembre 2013 – Compito

Tempo a disposizione: 3 ore

La soluzione comprende la **consegna elettronica** dei seguenti file mediante l'apposito applicativo Web **esamix** (<http://esamix.labx>):

GruppoAcquisto.zip file zip contenente sorgente java/class e pagine Web per punto 1
Voli.zip file zip contenente sorgente java/class, file xml e xsd per punto 2
Gite.zip file zip contenente sorgente java/class e file **persistenza.txt** per punto 3

Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, descrittori, risorse statiche o dinamiche, codice Java e relativi .class, ecc.) e NON dell'intero progetto

N.B. Per superare la prova scritta di laboratorio ed essere ammessi all'orale, è necessario totalizzare almeno 18 punti (su un totale disponibile di 33), equamente distribuiti sui tre esercizi, ovvero almeno 6 punti sul primo esercizio, 6 punti sul secondo esercizio e 6 punti sul terzo esercizio

Studenti in debito di Tecnologie Web L-A

Viene richiesto lo svolgimento dei soli esercizi 1 (17 punti) e 2 (16 punti). Tempo a disposizione: 2 ore.

I 18 punti necessari per l'ammissione all'orale sono così distribuiti: almeno 10 punti sul primo esercizio e almeno 8 punti sul secondo

ESERCIZIO 1 (11 punti)

Si realizzi un'**applicazione Web** che fornisca il servizio di acquisto di prodotti ortofrutticoli specificamente pensato per gruppi di acquisto, basata su **tecnologie Java servlet e JSP**.

In particolare, l'applicazione dovrà avere una pagina iniziale **ortofrutta.jsp** che autentichi l'utente con username e password (solo ed esclusivamente la prima volta all'interno di una sessione attiva) e che gli permetta di acquistare prodotti ortofrutticoli specificandone tipologia (zucchine, pesche, ...) e quantità (numero reale, in Kg). Tali acquisti personali devono essere mantenuti nel carrello utente.

In ogni istante sarà inoltre possibile, per un qualsiasi utente, svolgere il ruolo di coordinatore del gruppo di acquisto (per semplicità si supponga senza particolari credenziali di accesso): accedendo alla pagina **coordinatore.jsp** si potrà comandare il merge dei carrelli degli utenti desiderati (obbligatoriamente includendo il proprio e specificando il nome degli utenti coinvolti), creando così un "carrello di gruppo"; lo scopo è quello di poter poi effettuare un ordine cumulativo per grandi quantità di prodotti al fine di strappare prezzi migliori. Il calcolo (somma) delle quantità per ogni tipologia di prodotto dovrà essere effettuato da una servlet apposita. Quando il coordinatore finalizzerà il carrello di gruppo, tutti i carrelli degli utenti del gruppo dovranno essere invalidati/svuotati; ovviamente non deve essere possibile per più utenti svolgere il ruolo di coordinatore in concorrenza.

Tecnologie Web T

11 Settembre 2013 – Compito

ESERCIZIO 2 (11 punti)

Si progetti un **XML Schema** e un **documento XML** di esempio per la descrizione di un archivio personale di prenotazioni di voli aerei, rispettando le seguenti specifiche:

- ciascun documento XML modella un archivio personale;
- l'archivio personale è costituito da nome, cognome e numero passaporto della persona associata all'archivio e da zero o più prenotazioni;
- una prenotazione è rappresentata da: codice di prenotazione, nome della compagnia aerea, tipo di volo, aeroporto di partenza, data e ora di partenza, aeroporto di arrivo, data e ora di arrivo (tutti attributi obbligatori);
- il tipo di volo può essere: “Andata-Ritorno” o “Solo-Andata”;
- la data è formata dai campi “giorno”, “mese” e “anno”, mentre l'ora di partenza da “ora” e “minuti”; tutti i campi sono di tipo numero intero.

Si realizzi poi l'**applicazione Java “Voli”** che, facendo uso del **parser SAX**, esponga il metodo `getVoli()` riportato di seguito, unitamente ad un `main` di prova.

```
Set<String> getVoli();  
    restituisce l'insieme dei codici di prenotazione relativi ai voli di tipo “Andata-Ritorno”  
    operati da Ryanair di mattina (ovvero l'orario di volo è compreso tra le 5:00 e le  
    12:00), durante i mesi estivi (ovvero il mese di volo è compreso tra “Giungo” e “Ago-  
    sto”), e con partenza da “Bologna” o “Roma” o “Milano”
```

ESERCIZIO 3 (11 punti)

Con riferimento al seguente schema relazionale:

```
Studente(CodStud, Cognome, Nome, NickName)  
Gita(CodGita, Luogo, Data, Costo)  
Partecipazione(Studente, Gita)  
    FOREIGN KEY Studente REFERENCES Studente  
    FOREIGN KEY Gita REFERENCES Gita
```

si realizzi l'**applicazione Java-JDBC “Gite”** che esponga i **metodi CRUD per la gestione della persistenza** dei dati in esso contenuti, unitamente ad una classe `main` di prova per gli stessi.

Nel dettaglio, si richiede:

- la creazione delle tabelle all'interno del proprio schema nel database **TW_STUD** (esplicitando gli opportuni **vincoli di PK e FK**);
- l'implementazione dei metodi per il popolamento delle tabelle al punto sopra (passando i dati di input come parametri dei metodi)
 - inserire due o più studenti nella tabella “Studente” e due o più gite nella tabella “Gita”;
 - inserire alcune tuple di partecipazione nella tabella “Partecipazione”, previa verifica preventiva del rispetto dei vincoli di **FK**.
In caso di esito negativo (anche solo di uno dei due vincoli), il metodo deve rifiutare l'operazione di persistenza richiesta producendo un opportuno messaggio di errore sul file **persistenza.txt**;
- l'implementazione del metodo per il recupero dei *NickName degli studenti che hanno partecipato a più di 2 gite in Agosto spendendo meno di € 150* (stampare il risultato sul file **persistenza.txt**);
- l'implementazione del metodo per la cancellazione di *tutte le gite con destinazione “Roma”, unitamente alle tuple di partecipazione degli studenti ad esse collegate* (stampare il risultato sul file **persistenza.txt**).

N.B. L'implementazione deve limitarsi al solo **DBMS DB2**